


# Unidade VI:

## Ordenação Interna - Algoritmo da Bolha



**PUC Minas**

Adaptação dos slides elaborados pelo Instituto de Ciências Exatas e  
Informática - Departamento de Ciência da Computação

- **Funcionamento básico** 
- Algoritmo
- Análise do número de movimentações e comparações

- Técnica básica;
- Comparam-se dois elementos e trocam-se suas posições se o segundo elemento é menor do que o primeiro;
- São feitas várias passagens pelo vetor;
- Em cada passagem, comparam-se dois elementos adjacentes;
- Se estes elementos estiverem fora de ordem, eles são trocados;

# Exemplo

Legenda: - menor elemento em vermelho  
- parte ordenada está de azul

101 115 30 63 47 20

101    115    30    63    47    20

Bolha

# Exemplo

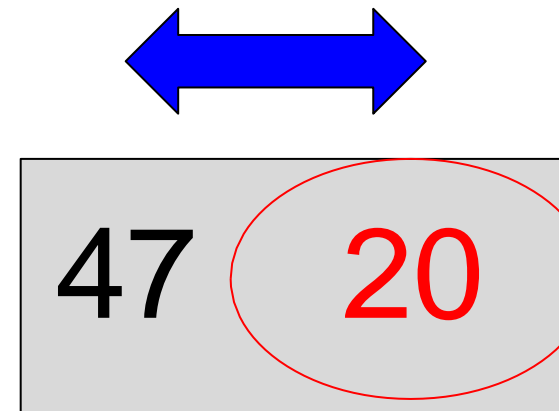
101    115    30    63

Comparação



Bolha

101      115      30      63

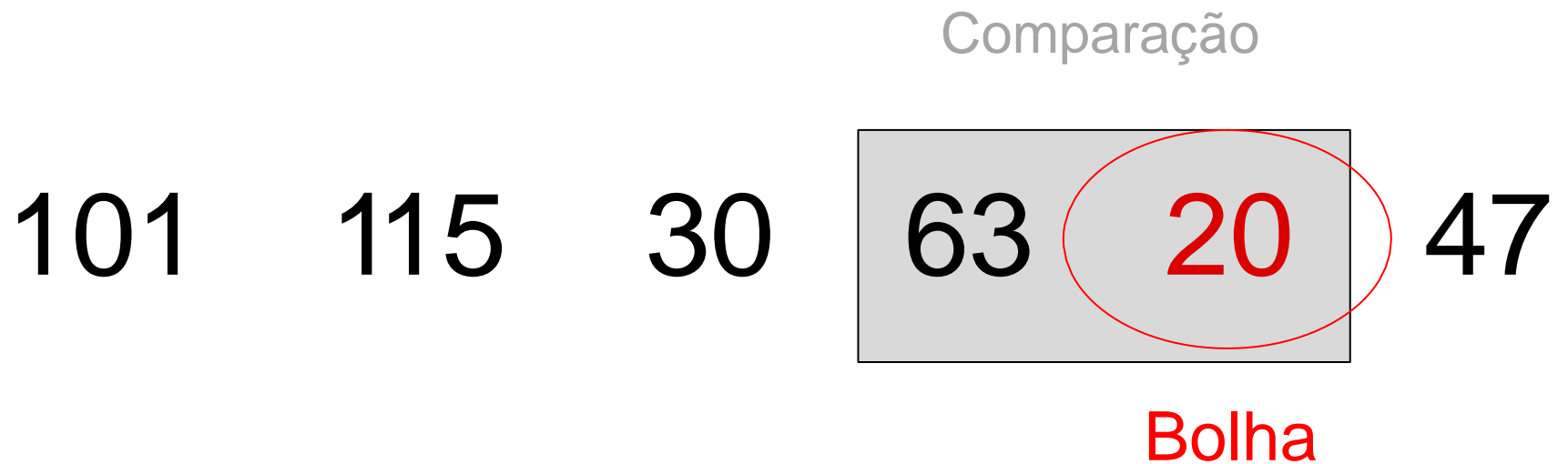


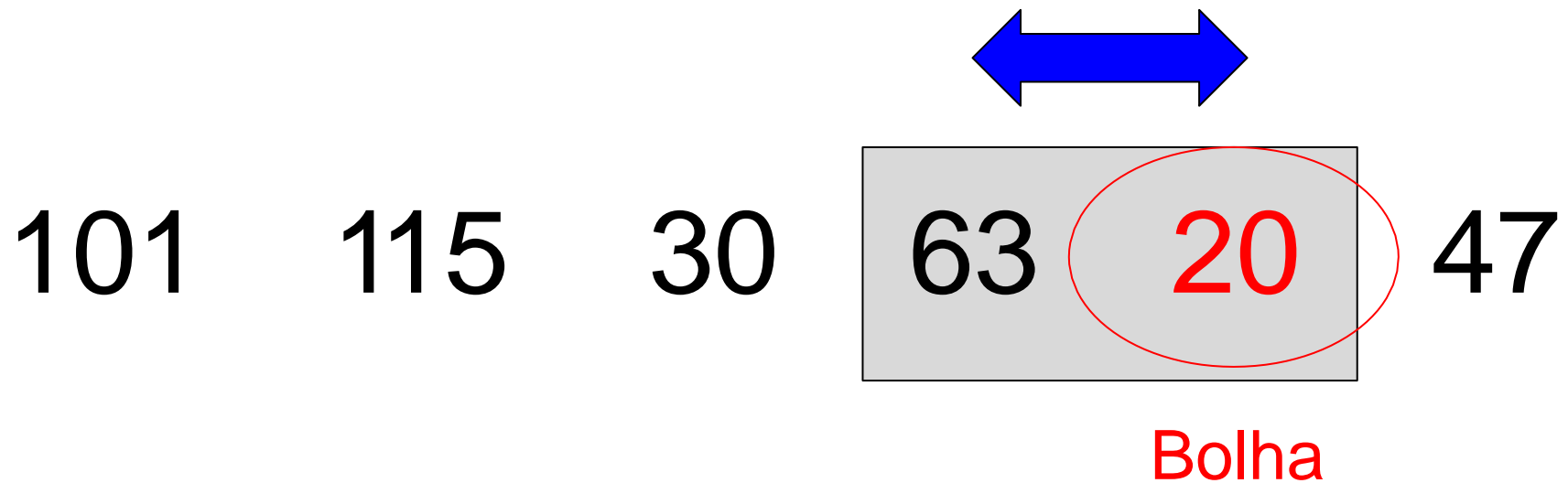
Bolha

101    115    30    63    20    47

Bolha

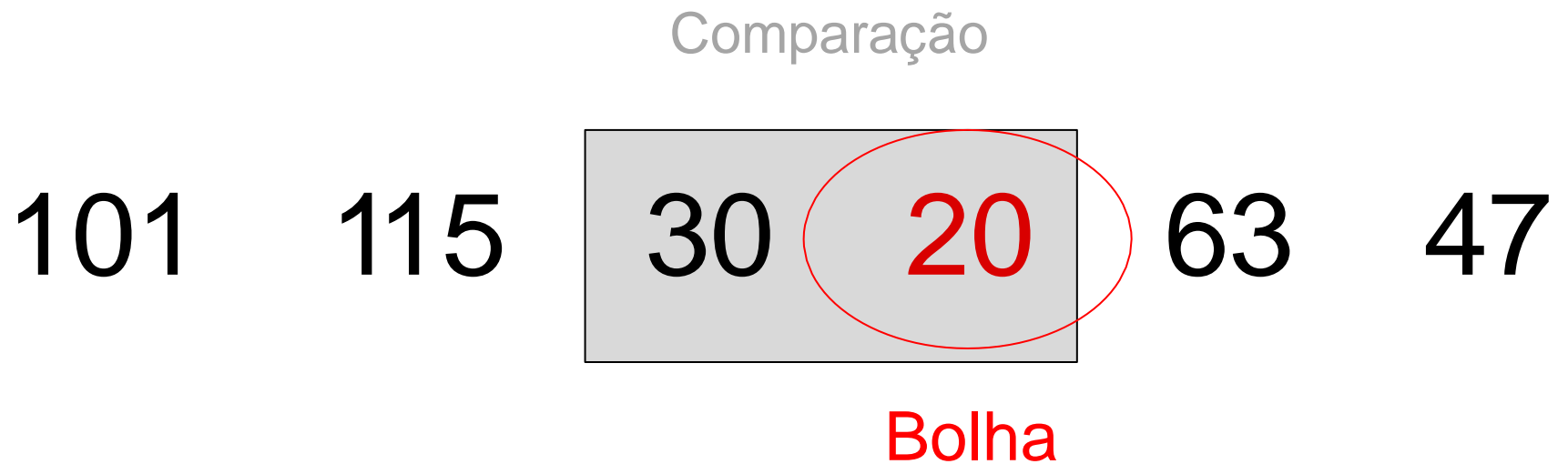


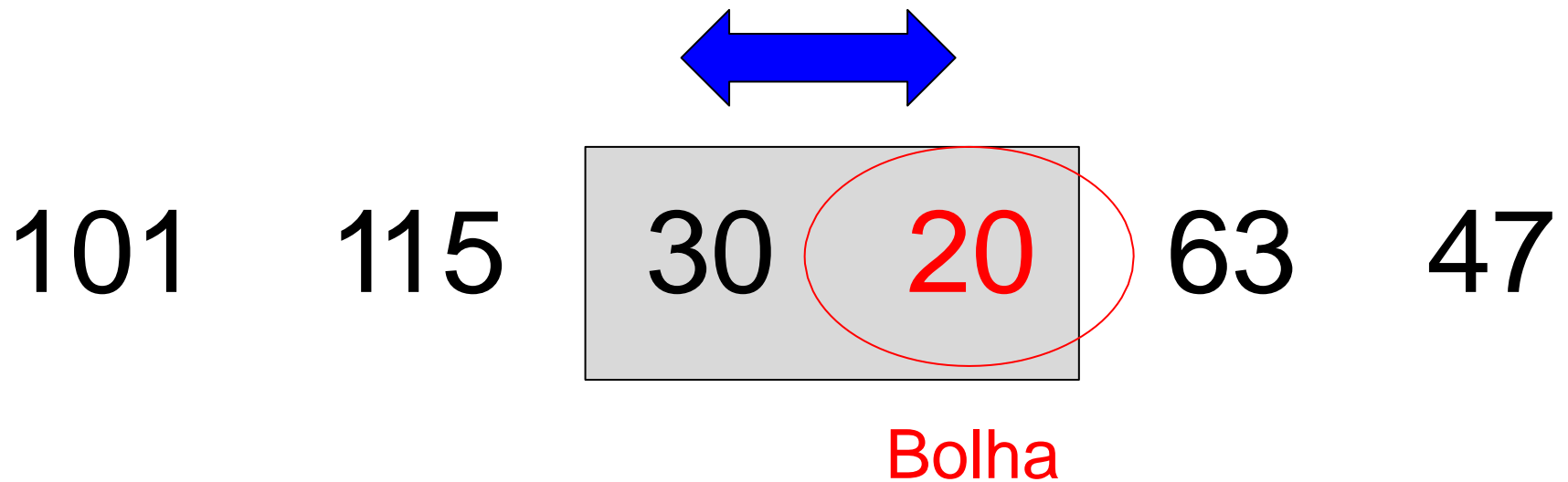




101    115    30    20    63    47

Bolha





101    115    20    30    63    47



Bolha

Comparação

101

115

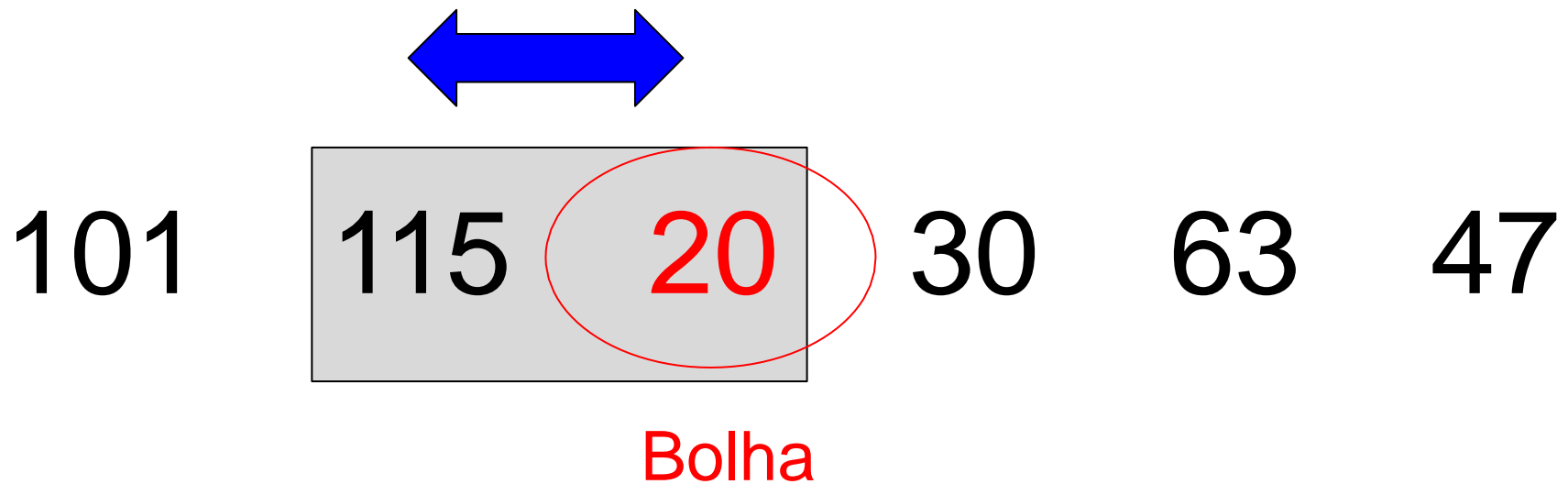
20

30

63

47

Bolha





101 20 115 30 63 47

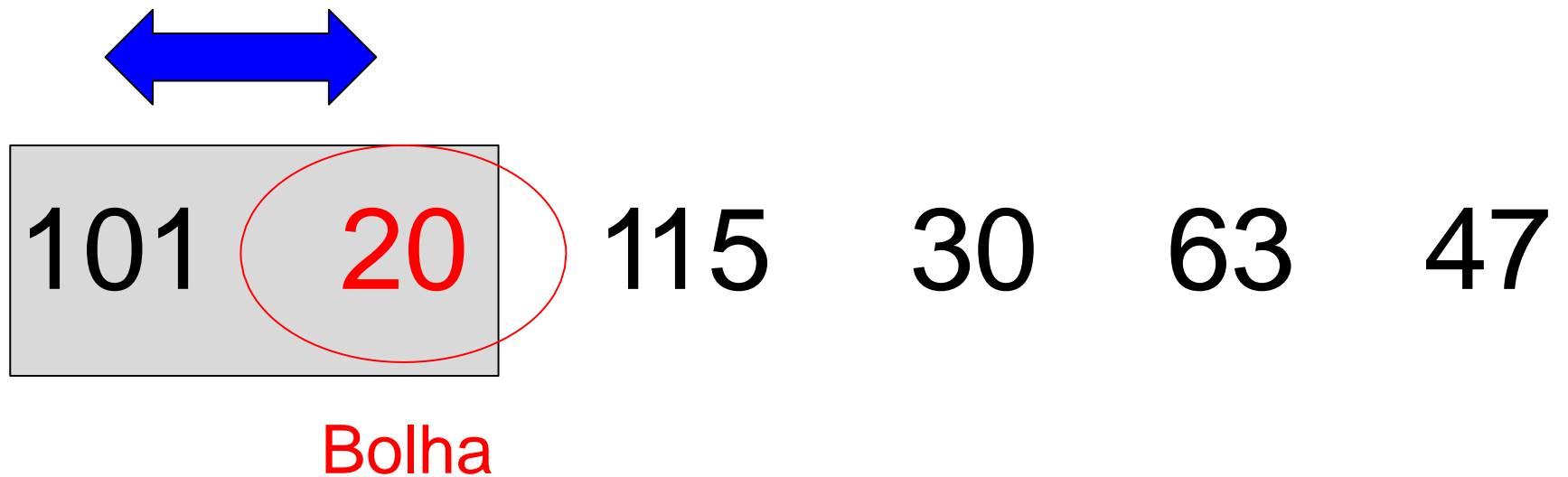


Bolha

Comparação



Bolha



20 101 115 30 63 47

Bolha

20 101 115 30 63 47

Ordenado

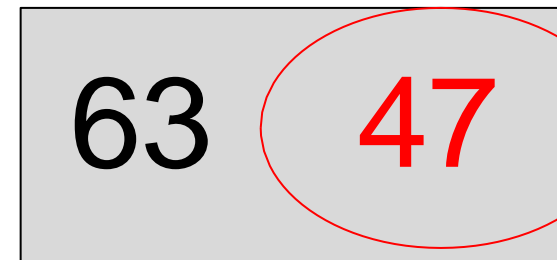
20 101 115 30 63 47

Bolha

# Exemplo

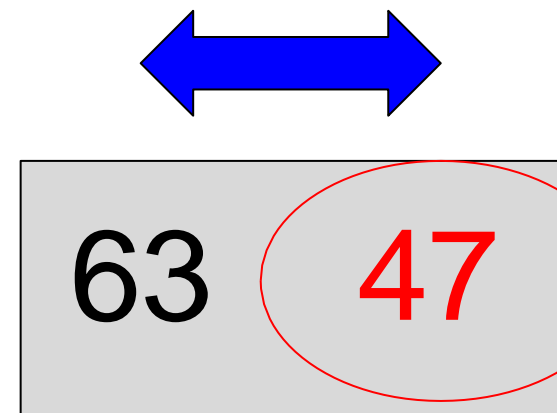
20 101 115 30

Comparação



Bolha

20 101 115 30

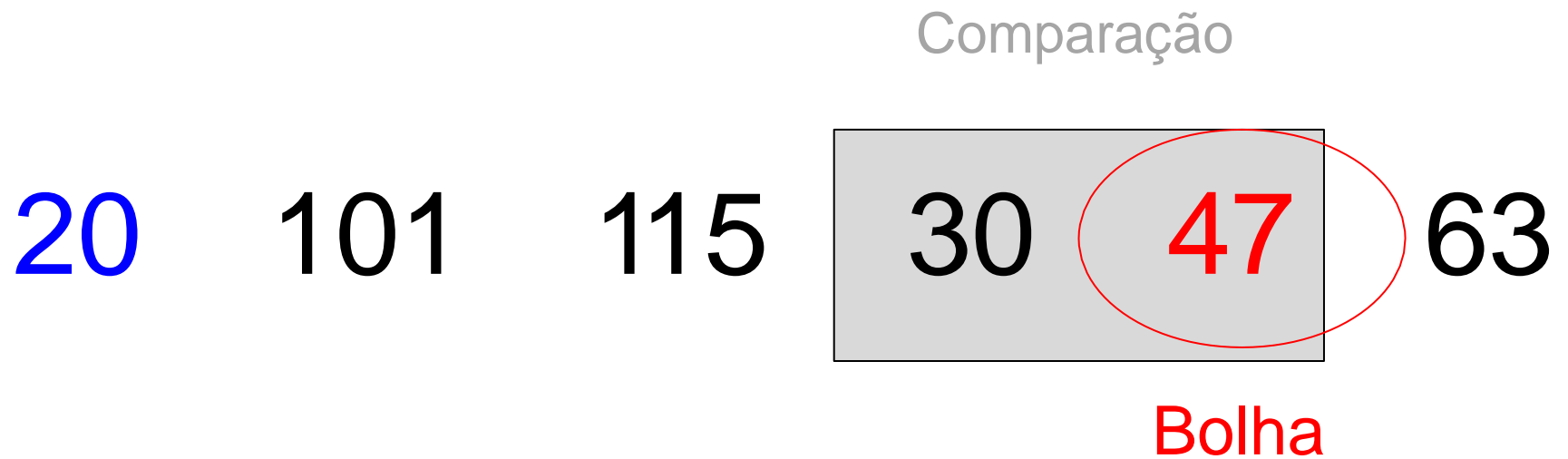


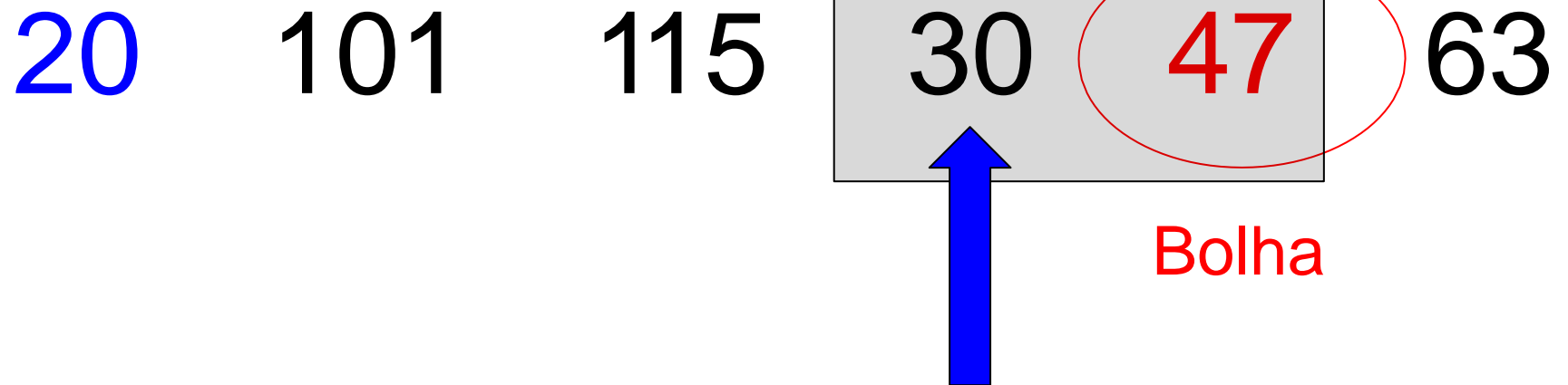
Bolha



20 101 115 30 47 63

Bolha



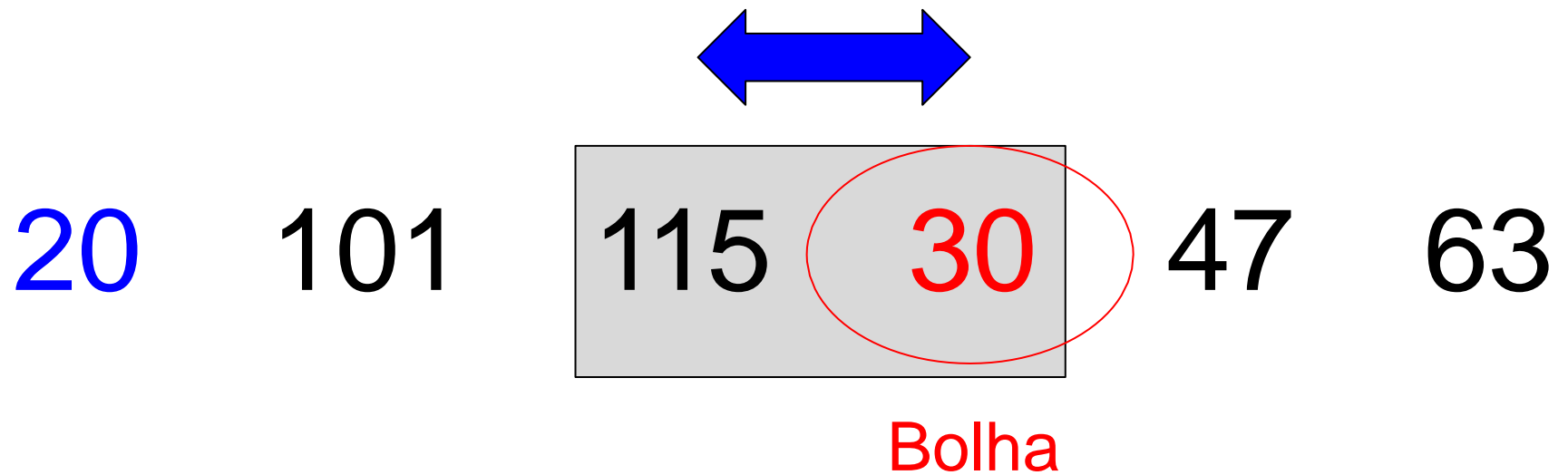


Menor (Será o número da bolha)

20 101 115 30 47 63

Bolha





20    101    30    115    47    63



Bolha

Comparação

20

101

30

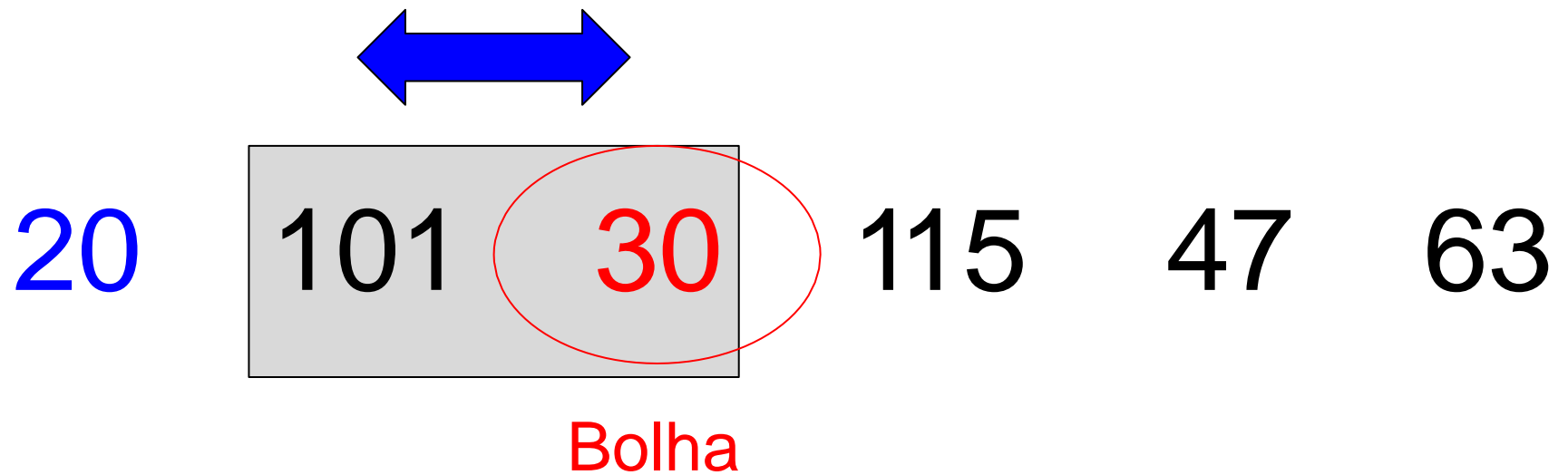
115

47

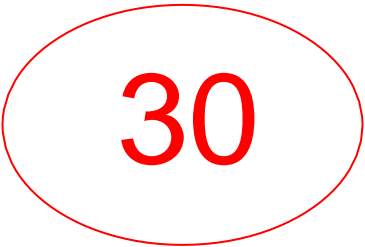
63

Bolha





20 30 101 115 47 63



Bolha

20 30 101 115 47 63

Ordenado

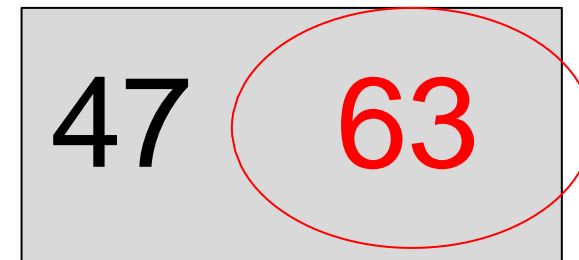
20 30 101 115 47 63

Bolha

# Exemplo

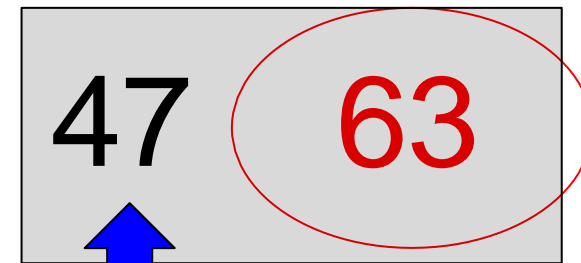
20 30 101 115

Comparação



Bolha

20 30 101 115

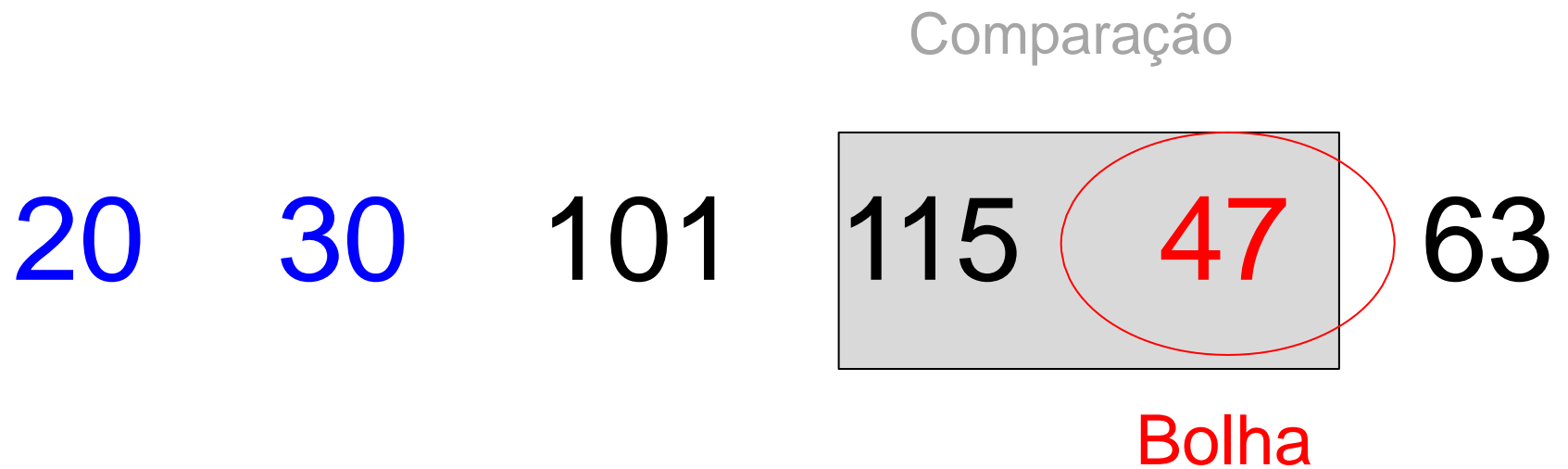


Bolha

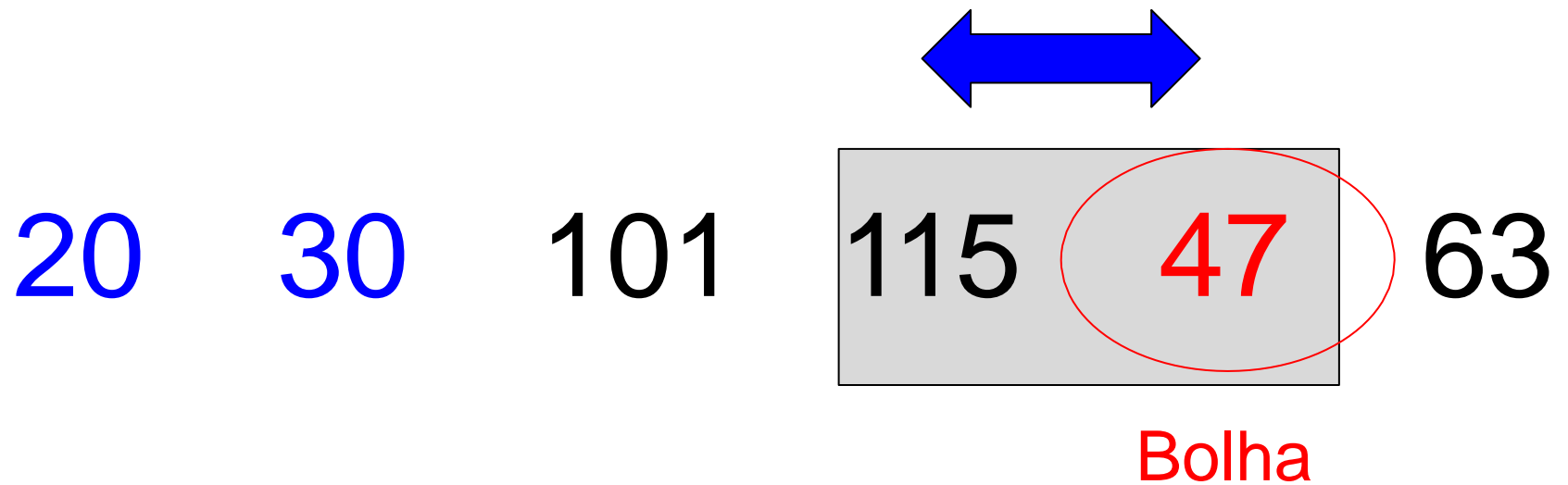
Menor (Será o número da bolha)

20 30 101 115 47 63

Bolha



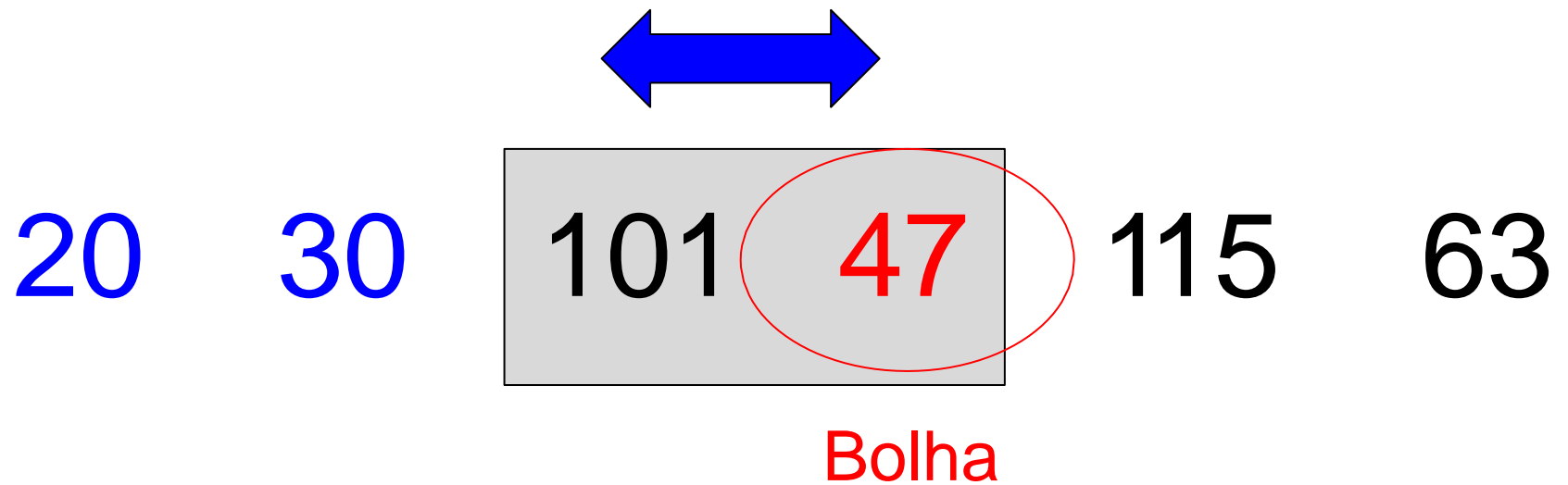





20 30 101 47 115 63

Bolha





20 30 47 101 115 63

A red oval highlights the number 47, which is positioned between 30 and 101. This indicates a comparison or swap operation in the bubble sort algorithm.

Bolha

20   30   47   101   115   63

Ordenado

20 30 47 101 115 63

Bolha

20 30 47 101

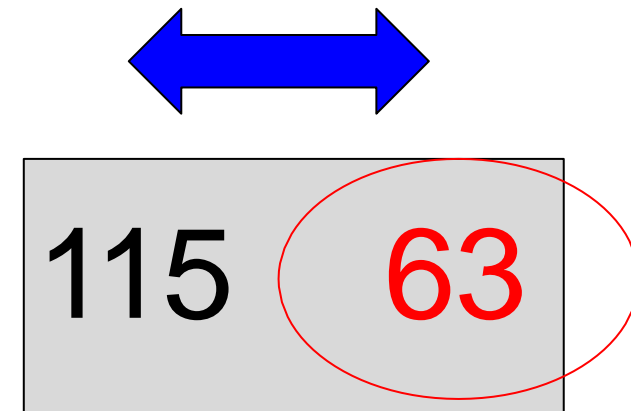
Comparação



Bolha



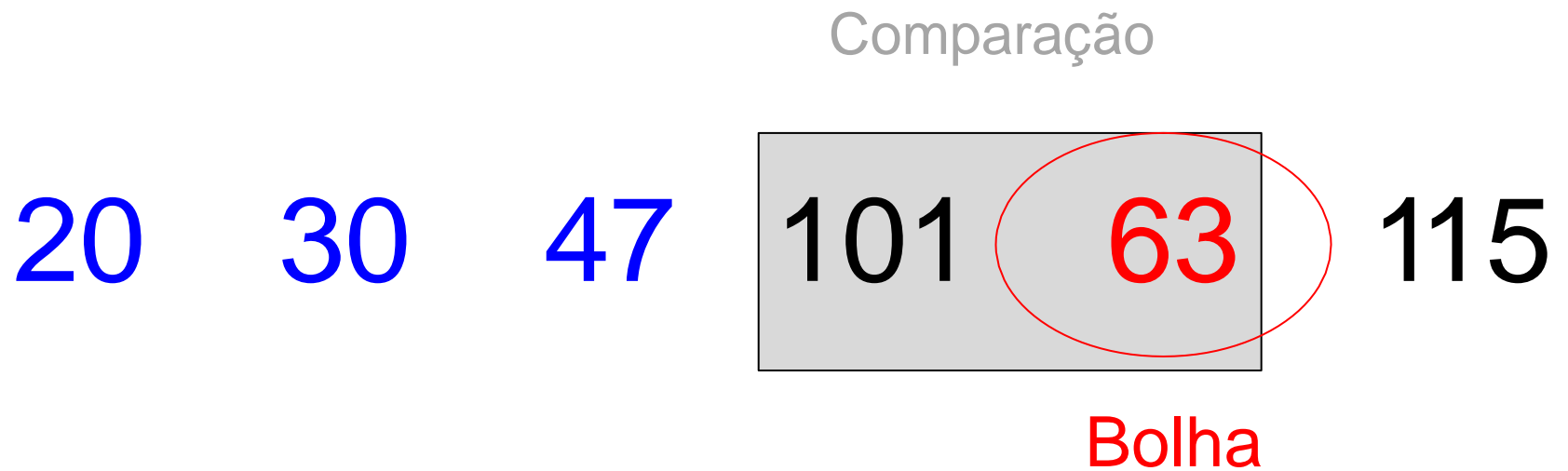
20 30 47 101

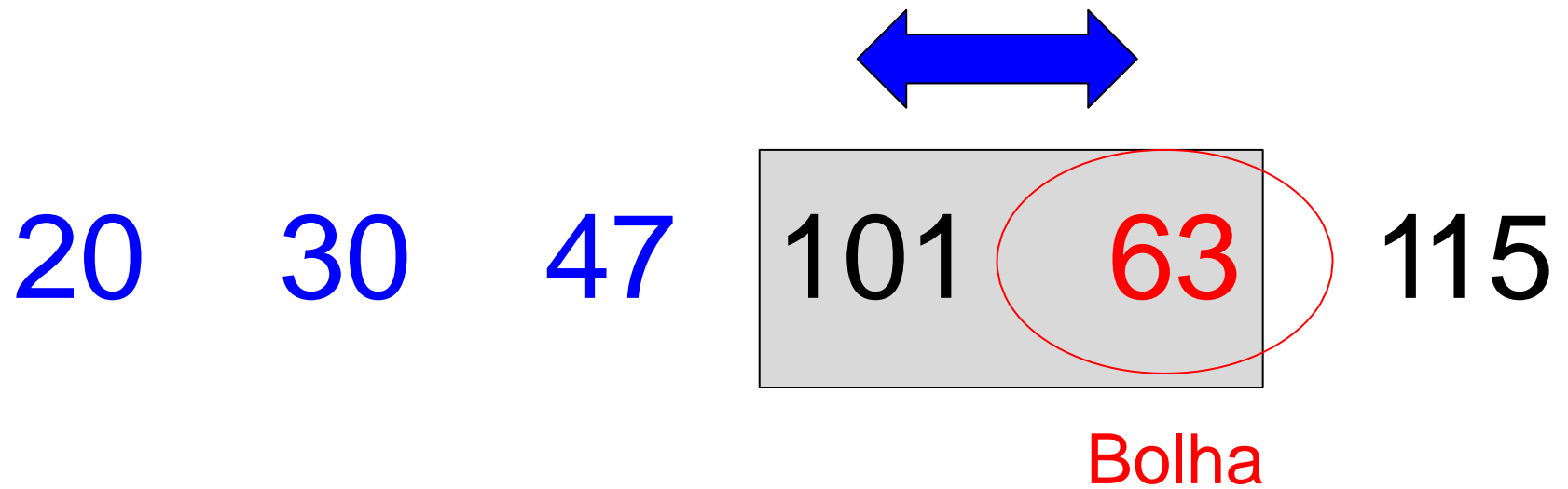


Bolha

20 30 47 101 63 115

Bolha





20 30 47 63 101 115



Bolha

20 30 47 63 101 115

Ordenado

20 30 47 63 101 115

Bolha

20 30 47 63

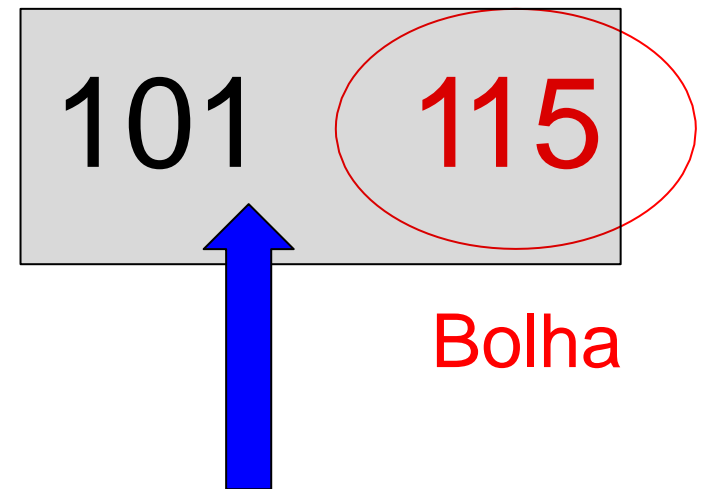
Comparação



Bolha



20 30 47 63




Menor (Será o número da bolha)

20 30 47 63 101 115

Bolha

20 30 47 63 101 115

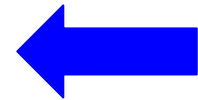
Ordenado

- Funcionamento básico
- **Algoritmo** 
- Análise do número de movimentações e comparações

## Algoritmo

```
int temp;  
for (int i = 0; i < (n - 1); i++) {  
    for (int j = n-1; j > i; j--){  
        if (array[j] < array[j-1]){  
            temp = array[j];  
            array[j] = array[j-1];  
            array[j-1] = temp;  
        }  
    }  
}
```

- Funcionamento básico
- Algoritmo
- **Análise do número de movimentações e comparações**



# Análise do Número de Comparações

```
int temp;  
for (int i = 0; i < (n - 1); i++) {  
    for (int j = n-1; j > i; j--){  
        if (array[j] < array[j-1]){  
            temp = array[j];  
            array[j] = array[j-1];  
            array[j-1] = temp;  
        }  
    }  
}
```

Exemplo:  $n = 5$

Para  $i = 0$ , os valores de  $j$  serão 4,3,2,1 = 4 vezes

Para  $i = 1$ , os valores de  $j$  serão 4,3,2 = 3 vezes

Para  $i = 2$ , os valores de  $j$  serão 4,3 = 2 vezes

Para  $i = 3$ , o valor de  $j$  será 4 = 1 vez

# Análise do Número de Comparações

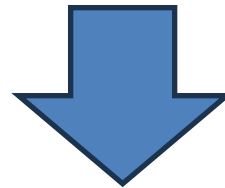
Exemplo:  $n = 5$

Para  $i = 0$ , os valores de  $j$  serão 4,3,2,1 = 4 vezes =  $(n-1)$

Para  $i = 1$ , os valores de  $j$  serão 4,3,2 = 3 vezes

Para  $i = 2$ , os valores de  $j$  serão 4,3 = 2 vezes

Para  $i = 3$ , o valor de  $j$  será 4 = 1 vez



$$1 + 2 + 3 + \dots + (n-1) =$$

$$\sum_{k=1}^{n-1} k$$



# Análise do Número de Comparações

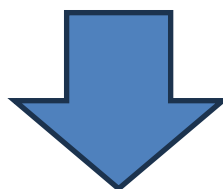
Exemplo:  $n = 5$

Para  $i = 0$ , os valores de  $j$  serão 4,3,2,1 = 4 vezes =  $(n-1)$

Para  $i = 1$ , os valores de  $j$  serão 4,3,2 = 3 vezes

Para  $i = 2$ , os valores de  $j$  serão 4,3 = 2 vezes

Para  $i = 3$ , o valor de  $j$  será 4 = 1 vez



$$1 + 2 + 3 + \dots + (n-1) = \sum_{k=1}^{n-1} k = \frac{n(n-1)}{2} = \frac{n^2}{2} - \frac{n}{2} = \Theta(n^2)$$

# Análise do Número de Movimentações

```
int temp;  
for (int i = 0; i < (n - 1); i++) {  
    for (int j = n-1; j > i; j--){  
        if (array[j] < array[j-1]){  
            temp = array[j];  
            array[j] = array[j-1];  
            array[j-1] = temp;  
        }  
    }  
}
```

# Análise do Número de Movimentações

```
int temp;  
for (int i = 0; i < (n - 1); i++) {  
    for (int j = n-1; j > i; j--){  
        if (array[j] < array[j-1]){  
            temp = array[j];  
            array[j] = array[j-1];  
            array[j-1] = temp;  
        }  
    }  
}
```

- Pior caso: o *array* está ordenado de forma decrescente (if sempre é verdadeiro)

$$M_{Max}(n) = 3 * C(n)$$

Número de  
movimentações

# Análise do Número de Movimentações

```
int temp;  
for (int i = 0; i < (n - 1); i++) {  
    for (int j = n-1; j > i; j--){  
        if (array[j] < array[j-1]){  
            temp = array[j];  
            array[j] = array[j-1];  
            array[j-1] = temp;  
        }  
    }  
}
```

- Pior caso: o *array* está ordenado de forma decrescente (if sempre é verdadeiro)

$$M_{Max}(n) = 3 * C(n)$$



$$M_{Max}(n) = 3 * \frac{n(n-1)}{2}$$

# Análise do Número de Movimentações

```
int temp;  
for (int i = 0; i < (n - 1); i++) {  
    for (int j = n-1; j > i; j--){  
        if (array[j] < array[j-1]){  
            temp = array[j];  
            array[j] = array[j-1];  
            array[j-1] = temp;  
        }  
    }  
}
```

- Melhor caso: o *array* está ordenado de forma crescente, assim nenhuma troca será feita (if sempre é falso)  $\rightarrow O(1)$

# Conclusão

- Vantagens:
  - Algoritmo simples;
  - Algoritmo estável;
  - *In Place*;
  -

## • Desvantagens

- 
- O problema dos algoritmos de seleção e da bolha é porque eles realizam várias comparações redundantes, não adaptável;
- O algoritmo faz muitas trocas (número quadrático de movimentações)
- Sendo assim, possui complexidade em termos de comparações constante, sendo  $O(n^2)$ , tanto para o pior caso, caso médio e melhor caso;